

APPENDIX

[Contents](#) [Previous](#) [Next](#)

APPENDIX A

CDMS Classes

Figure 1, "CDMS Classes", on page 175 illustrates the class inheritance structure of CDMS. The classes may be categorized as abstract or concrete. Only concrete classes are meant to be used directly. In contrast an abstract class defines the common interface of its subclasses. For example, the class *AbstractAxis2D* defines the common interface for two-dimensional coordinate axes. It has concrete subclasses *DatasetAxis2D*, *FileAxis2D*, and *TransientAxis2D*, which are used in applications. Abstract classes are denoted in italics.

For many abstract classes there are three 'flavors' of subclass: dataset, file, and transient. Dataset-related objects are thought of as being contained in datasets in the sense that operations on those objects result in I/O operations on the corresponding dataset. The same is true of file-related objects. Objects in datasets and files are examples of persistent objects, whose state persists after the application exits. On the other hand, transient objects live in memory and are not persistent.

In general the concrete subclasses closely mirror the interface of the abstract parent class. For this reason this document defines the interfaces of the abstract classes, and only discusses a concrete class in the few cases where the interface has been extended. This allows applications to treat the behavior of, say a dataset axis and file axis, as identical.

FIGURE 1. CDMS Classes

APPENDIX B

Version Notes

B.1 Version 4.0

CDMS version 4.0 adds support for nonrectangular grids:

1. The following grid classes were added: *AbstractHorizontalGrid*, *AbstractCurve-Grid*, *AbstractGenericGrid*, *DatasetCurveGrid*, *FileCurveGrid*, *TransientCurve-Grid*, *DatasetGenericGrid*, *FileGenericGrid*, and *TransientGenericGrid*.
2. The following axis classes were added: *AbstractCoordinateAxis*, *AbstractAuxAxis1D*, *AbstractAxis2D*, *DatasetAuxAxis1D*, *FileAuxAxis1D*, *TransientAuxAxis1D*, *DatasetAxis2D*, *FileAxis2D*, and *TransientAxis2D*.
3. The *getMesh* and *clone* methods were added for grids.
4. An interface to the SCRIP package was added.

B.2 Version 3.0 Overview

CDMS version 3.0 is a significant enhancement of previous versions. The major changes were:

1. CDAT/CDMS was integrated with the Numerical Python masked array class MA.MaskedVariable. The MV submodule was added as a wrapper around MA.
2. Methods that read data, such as subRegion, subSlice, and the slice operations, return instances of class TransientVariable. The plot and regrid modules were modified to handle masked array input. The specifiers time=..., latitude=..., etc. were added to the I/O routines.
3. The class TransientVariable was added.
4. A number of new functions were added, notably subRegion and subSlice, which return instances of TransientVariable.
5. When a masked array is returned from a method, it is "squeezed": singleton dimensions are removed. In contrast, transient variables are not squeezed. I/O functions have a *squeeze* option. The method setAutoReshapeMode was removed.
6. Internal attributes are handled in the InternalAttributes class. This allows CDMS classes to be subclassed more readily.
7. The class Variable was renamed DatasetVariable.
8. The cu module was emulated in cdms. cu and cdms methods can be mixed.
9. The code was modularized, so that Python, CDMS, and Numerical Python can be built and installed separately. This significantly enhances the portability of the code.

B.3 V3.0 Details

B.3.1 AbstractVariable

- Functions getDomain, getSlice, rank, regrid, setMissing, size, subRegion, and subSlice were added.
- The functions getRegion, getSlice, getValue, and the slice operators all return an instance of MA, a masked array. Singleton dimensions are squeezed.
- The functions subRegion and subSlice return an instance of TransientVariable. Singleton dimensions are not squeezed.
- The xxSlice and xxRegion functions have keywords *time*, *level*, *latitude*, and *longitude*.
- The input functions have the keyword *squeeze*.
- AbstractVariable inherits from class Slab. The following functions previously available in module cu are Slab methods: getattribute, setattribute, listdimattributes, getdimattribute, listall, and info.
- AbstractVariable implements arithmetic functions, astype.
- The write function was added.

B.3.2 AbstractAxis

- The functions asComponentTime, asRelativeTime, clone, getAxisIds, getAxisIndex, getAxisList, getAxisListIndex, mapIntervalExt were added.
- subaxis was renamed subAxis for consistency.
- Generalized wraparound was implemented, to handle multiple cycles, reversing, and negative strides. By default, coordinate intervals are closed. The intersection options 'n','e','b',and 's' were added to the interval indicator – see mapIntervalExt.

B.3.3 AbstractDatabase

- The function open is synonymous with openDataset.

B.3.4 Dataset

- The function open is synonymous with openDataset.

B.3.5 cdms module

- The functions `asVariable`, `isVariable`, and `createVariable` were added.
- The function `setAutoReshapeMode` was removed. It is replaced by the `squeeze` option for all I/O functions.

B.3.6 CdmsFile

- The function `createVariable` has a keyword `fill_value`. The datatype may be a Numeric/MA typecode.
- The function `write` was added.

B.3.7 CDMSError

- All errors are an instance of the class `CDMSError`.

B.3.8 AbstractRectGrid

- The function `createGaussianGrid` was added.

B.3.9 InternalAttributes

- The class `InternalAttributes` was added. It has methods `add_internal_attribute`, `is_internal_attribute`, and `replace_external_attributes`.

B.3.10 TransientVariable

- The class `TransientVariable` was added. It inherits from both `AbstractVariable` and `MA`.
- The `cdms` module function `createVariable` returns a transient variable.
- This class does not implement the functions `getPaths` or `getTemplate`.

B.3.11 MV

- The `MV` submodule of `cdms` was added.

APPENDIX C

cu Module

The `cu` module is the original CDAT I/O interface. As of version 3 it is emulated in the `cdms` module. It is maintained for backward compatibility.

The `cu` classes are `Slab`, corresponding to `TransientVariable` in CDMS, and `cuDataset`, corresponding to `Dataset` in CDMS.

C.1 Slab

Table C.1 Slab Methods

Type Definition

Various	<code>getattribute(name)</code> Get the value of an attribute. <code>name</code> is the string name of the attribute. The following special names can always be used: <code>'filename'</code> , <code>'comments'</code> , <code>'grid_name'</code> , <code>'grid_type'</code> , <code>'time_statistic'</code> , <code>'long_name'</code> , <code>'units'</code> .
---------	--

Various	<code>getdimattribute(dim, field)</code> Get the value of a dimension attribute. dim is the dimension number, an integer in the range 0..rank-1. field is a string, one of: "name", "values", "length", "units", "weights", "bounds".
None	<code>info(flag=None, device=sys.stdout)</code> Print slab information. If flag is nonzero, dimension values, weights, and bounds are also printed. Output is sent to device.
List	<code>listall(all=None)</code> Print slab information. If all is nonzero, dimension values, weights, and bounds are also printed.
List	<code>listdimattributes(dim, field)</code> List dimension attributes. Returns a list of string attribute names which can be input to <code>getdimattribute</code> . dim is the dimension number, an integer in the range 0..rank-1. field is a string, one of: "name", "values", "length", "units", "weights", "bounds".
None	<code>setattribute(name, value)</code> Set an attribute. name is the string name of the attribute. value is the value of the attribute.

C.2 cuDataset

Table C.2 cuDataset Methods

Type Definition

None	<code>cleardefault()</code> Clear the default variable name.
None	<code>default_variable(vname)</code> Set the default variable name. vname is the string variable name.
Array	<code>dimensionarray(dname, vname=None)</code> Values of the axis named dname. dname is the string axis name. vname is the string variable name. The default is the variable name set by <code>default_variable</code> .
Axis	<code>dimensionobject(dname, vname=None)</code> Get an axis. dname is the string name of an axis. vname is a string variable name. The default is the variable name set by <code>default_variable</code> .
Various	<code>getattribute (vname, attribute)</code> Get an attribute value. vname is a string variable name. attribute is the string attribute name.

String	<p>getdimensionunits (dname,vname=None)</p> <p>Get the units for the given dimension.</p> <p>dname is the string name of an axis.</p> <p>vname is a string variable name. The default is the variable name set by default_variable.</p>
Various	<p>getglobal (attribute)</p> <p>Get the value of the global attribute.</p> <p>attribute is the string attribute name.</p>
Variable	<p>getslab (vname, *args)</p> <p>Read data for a variable.</p> <p>vname is the string name of the variable.</p> <p>args is an argument list corresponding to the dimensions of the variable. Arguments for each dimension can be:</p> <p>(1) ':' or None -- select the entire dimension</p> <p>(2) Ellipsis -- select entire dimensions between the ones given.</p> <p>(3) a pair of successive arguments giving an interval in world coordinates.</p> <p>(4) a CDMS-style tuple of world coordinates e.g. (start, stop, 'cc')</p>
List	<p>listall (vname=None, all=None)</p> <p>Get info about data from the file.</p> <p>vname is the string name of the variable.</p> <p>If all is non-zero, dimension values, weights, and bounds are returned as well</p>
List	<p>listattribute (vname=None)</p> <p>Return a list of attribute names.</p> <p>vname is the name of the variable. The default is the variable name set by default_variable.</p>
List	<p>listdimension (vname=None)</p> <p>Return a list of the dimension names associated with a variable.</p> <p>vname is the name of the variable. The default is the variable name set by default_variable.</p>
List	<p>listglobal ()</p> <p>Return a list of the global attribute names.</p>
List	<p>listvariable ()</p> <p>Return a list of the variables in the file.</p>
None	<p>showall (vname=None, all=None, device=sys.stdout)</p> <p>Print a description of the variable.</p> <p>vname is the string name of the variable.</p> <p>If all is non-zero, dimension values, weights, and bounds are returned as well.</p> <p>Output is sent to device.</p>
None	<p>showattribute (vname=None, device=sys.stdout)</p> <p>Print the attributes of a variable.</p> <p>vname is the string name of the variable.</p> <p>Output is sent to device.</p>
None	<p>showdimension (vname=None, device=sys.stdout)</p> <p>Print the dimension names associated with a variable.</p> <p>vname is the string name of the variable.</p>

	Output is sent to device.
None	showglobal (device=sys.stdout) Print the global file attributes. Output is sent to device.
None	showvariable (device=sys.stdout) Print the list of variables in the file.

[Contents](#) [Previous](#) [Next](#)